

Multiprojeção por Clusterização de Visualização de Ambientes Virtuais com o Xj3D

Eduardo de Lucena Falcão, Hermenegildo Remígio da Rocha Neto, Liliane dos Santos Machado
Laboratório de Tecnologias para o Ensino Virtual e Estatística
Universidade Federal da Paraíba - CCEN
{eduardolfalcao, hdrneto}@gmail.com, liliane@di.ufpb.br

Resumo

Sistemas de multiprojeção visam particionar a imagem de modo a oferecer diferentes configurações de visualização em mais de um monitor. O presente trabalho tem como objetivo apresentar uma solução de multiprojeção para visualização de ambientes virtuais tridimensionais no padrão X3D. Utilizando o browser Xj3D, de código aberto, em conjunto com a LibGlass, é apresentado o resultado obtido com a descrição das etapas de desenvolvimento.

Abstract

Multiprojection systems aims the partition of images to offer different visualization configurations into multiple displays. This work has as goal to present a solution for multiprojection of virtual environments developed following the X3D standard. Thus, the description of steps necessary is presented for the use of the Xj3D, an open source browser, with the LibGlass for visualization clustering.

1. Introdução

Com a constante evolução tecnológica, o uso de técnicas de Realidade Virtual (RV) tem aumentado expressivamente na comunidade científica. Conceitos relacionados à visualização tridimensional e interatividade, são utilizados para prover ao usuário ambientes virtuais tridimensionais com maior imersão. Estes ambientes virtuais podem atuar em diversas áreas, dando suporte a educação [7], entretenimento [6], treinamento [3], preservação do patrimônio histórico [8], disseminação cultural, entre outros.

No contexto de visualização, sistemas de multiprojeção podem ser considerados para prover o particionamento da representação com abordagens baseadas em hardware ou baseadas em software. Este trabalho tem por objetivo apresentar uma solução de multiprojeção através da integração entre o browser Xj3D, para o processamento de ambientes X3D, e a libGlass, para clusterização da visualização.

2. Multiprojeção

Um sistema de multiprojeção é um sistema computacional complexo para aplicações que utilizam RV de alta resolução, com múltiplas projeções - utilizando ou não estereoscopia - particionando um ambiente ou imagem, produzido em tempo real, através de várias telas. A multiprojeção pode ser alcançada de diversas formas, tanto através de *software* quanto *hardware* específico para tal finalidade [4]. O hardware utilizado em sistemas de multiprojeção abrange placas gráficas, PCs com alto poder computacional, monitores ou projetores. É necessária também a utilização de *software* para que seja realizado o cálculo da angulação e posição das telas. Um exemplo de solução de *software* para sistemas de multiprojeção é a libGlass[1], biblioteca para programação distribuída.

Em um sistema de multiprojeção como uma parede de multiprojeção (*PowerWall*) [10], as telas são distribuídas em forma de matriz (Figura 1).

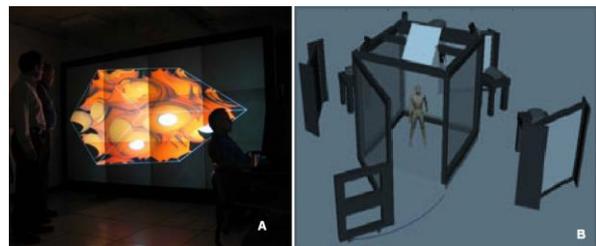


Figura 1 – A: imagem projetada através de matriz de displays 3x4. B: imagem ilustrativa da CAVERNA Digital. Fonte: [4], [10].

É possível ainda alcançar multiprojeção estendendo telas horizontalmente com angulação entre elas. Exemplos são sistemas como a CAVERNA™ Digital [4] ou NAVE [5]. No caso da CAVERNA Digital, é possível ter de quatro a seis projeções (cubo), com angulação de 90° entre elas, onde o usuário encontra-se totalmente imerso no ambiente (Figura 1). No sistema de multiprojeção NAVE a visualização é alcançada através de três telas horizontalmente dispostas lateralmente. A angulação necessária para um bom nível de imersão é de 120°.

3. X3D

O X3D (*Extensible 3D*) é um padrão adotado internacionalmente para disponibilizar conteúdos tridimensionais na Web [2]. É um padrão aberto que permite descrever em um arquivo formas e comportamentos de ambientes virtuais tridimensionais complexos e interativos.

Uma derivação do X3D em fase experimental é o X3DOM (*X-Freedom*), *framework* de código aberto que visa preencher a especificação do HTML5 promovendo a integração do mesmo com conteúdos 3D declarativos, como o X3D [12]. Como resultado é possível navegar em um ambiente X3D em navegadores Web sem necessidade de *plugins* ou *browsers* X3D específicos.

A visualização dos ambientes virtuais tridimensionais descritos em X3D é realizada através de *browsers* específicos capazes de interpretar e processar as cenas (arquivos X3D), apresentando os modelos tridimensionais e permitindo interações do usuário com os objetos. Os *browsers* X3D, comumente chamados de navegadores ou *players*, podem se apresentar como *plugins* em navegadores Web, como o *Google Chrome* ou *Mozilla Firefox*, ou como aplicações independentes.

Os *browsers* para X3D são implementados visando interpretar as funcionalidades definidas por sua especificação [11]. Assim, desenvolvedores destes *browsers* têm autonomia para seleção de funcionalidades a serem suportadas além das básicas, podendo também adicionar novas. Devido à característica de extensibilidade do X3D existem múltiplos documentos da especificação que conduzem à evolução coerente de capacidades diversas do X3D. Cada documento pode ser desenvolvido e integrado de forma independente, permitindo atualizações anuais e crescimento estável do formato. A produção destes documentos ocorre por meio de um processo livre e colaborativo de membros voluntários do grupo Web3D. Cada ano novas funcionalidades são propostas, avaliadas, especificadas formalmente e enviadas para a ISO (*International Organization for Standardization*), a fim de serem revisadas e confirmadas.

Pelas suas funcionalidades e capacidades de extensibilidade, o X3D também se apresenta como importante tecnologia para a produção de sistemas de realidade virtual imersivos, como é o caso da multiprojeção. Vários *browsers* X3D possuem suporte para tais sistemas, e outros disponibilizam seu código para que desenvolvedores sintam-se livre para implementá-los. A Tabela 1 exhibe as características dos principais *browsers* X3D.

Tabela 1 – *Browsers* X3D e suas características.

<i>Browser</i>	Uso	Código fonte e linguagem	Meio de disponibilização
Bs Contact	Pago*	Indisponível	Aplicação local, <i>plugin</i> , <i>applet</i> , disp. móveis, sist. de multiprojeção
FreeWRL	Gratuito	<i>Open-source</i> C	Aplicação local, <i>plugin</i>
Instant Player	Gratuito **	Indisponível	Aplicação local, <i>plugin</i> , disp. móveis, sist. de multiprojeção
OpenVrml	Gratuito	<i>Open-source</i> C++	Aplicação local, <i>plugin</i>
Xj3D	Gratuito	<i>Open-source</i> Java	Aplicação local, <i>Applet</i> , <i>Java Web Start</i>

*Existe uma versão gratuita experimental

**Necessita de permissão para uso comercial

O Xj3D é um *browser* em conformidade com as especificações do X3D, implementando inclusive funcionalidades não especificadas pelo padrão, o que facilita o desenvolvimento de ambientes virtuais 3D com interatividade complexa e alta qualidade gráfica. Nenhum dos *browsers* gratuitos e de código aberto possui suporte a multiprojeção, e dentre estes o Xj3D é o que possui mais recursos disponíveis. Estes fatores resultaram na escolha do *browser* Xj3D para a implementação da multiprojeção.

4. LibGlass

Um dos principais requisitos em sistemas distribuídos é a eficiência de sincronização dos computadores e subsistemas. Uma das formas de sincronização utilizada é através de redes de computadores, que implica na necessidade de troca de mensagens entre os computadores da rede com um custo muito menor, graças ao uso de componentes convencionais nesta abordagem.

A LibGlass (ou Glass) é uma biblioteca para computação distribuída, focando em alta performance de aplicações de RV. A Glass é um conjunto de componentes escaláveis que facilita a reprogramação de aplicações não-distribuídas a fim de torná-las distribuídas. Com a Glass esta reescrita pode ser feita em parte da aplicação, ou em uma nova [1].

Todas as funcionalidades da Glass são providas pelos *plugins* que são acessados através de simples declaração de variável. Atualmente existem quatro *plugins* disponibilizados pela Glass: variáveis compartilhadas, barreiras, eventos e *aliases*. As variáveis compartilhadas são úteis para dados de tamanho pequeno, sincronizadas (evitando problemas

de acesso global e inconsistência) e independem de um tipo específico. As barreiras servem para evitar o acesso concorrente a um mesmo dado. Eventos são basicamente filas de valores, úteis para a propagação de mudanças assíncronas, e são independentes do tipo. Por último, os *aliases* são úteis para calcular de forma transparente o ângulo de visão de cada projeção de uma aplicação com multiprojeção, e definem a posição da câmera segundo esse ângulo.

A Glass trabalha utilizando o padrão Cliente-Servidor. O servidor é instanciado para que os clientes (ou nós) conectem-se a ele. O primeiro cliente que se conectar ao servidor será inicializado como mestre e os próximos clientes serão considerados escravos. A proposta da LibGlass é utilizar um protocolo de transporte, como o TCP, para auxiliar na troca de mensagens e sincronização dos nós do sistema.

5. Desenvolvimento

Existem várias linguagens e *frameworks* para descrição de ambientes virtuais tridimensionais. A LibGlass pode ser uma solução para a implementação de sistemas de multiprojeção, nos *softwares* de código aberto que executam tais ambientes. Neste trabalho o padrão de descrição tridimensional escolhido é o X3D, e como programa que processa o arquivo X3D disponibilizando a cena correspondente, foi adotado o Xj3D devido às características descritas na seção 3.

Para particionar a visualização do ambiente virtual em múltiplas telas foi necessário compreender em profundidade a forma de renderização do Xj3D. Sua implementação disponibiliza quatro etapas para a configuração do processo de renderização da cena:

- a primeira etapa trata a pré-configuração das estruturas renderizáveis da cena, funcionando como uma espécie de interface entre o grafo de cena em alto nível e uma camada de gerenciamento da OpenGL em baixo nível. Esta etapa ocorre uma única vez durante a execução do *browser*;
- a segunda consiste em qualquer configuração prévia à renderização do grafo de cena, como por exemplo, o modo de preenchimento e desenho dos polígonos;
- a terceira etapa é a renderização, onde o grafo de cena será desenhado em tela de acordo com parâmetros configurados nas etapas anteriores;
- na última etapa podem ser efetuadas configurações posteriores à renderização. Após esta fase, a segunda etapa é executada repetindo o ciclo.

Como na segunda fase ocorre toda pré-configuração imediata à renderização da cena, nesta

etapa foi implementada a distribuição da visualização para alcançar maior agilidade na definição da localização da câmera. Para tal, foi necessária a criação de métodos de cálculo das matrizes de posição e de orientação da câmera, para possibilitar o compartilhamento da visão com o auxílio da LibGlass. Mudanças adicionais foram feitas nas camadas de configuração de cena (primeira etapa) e renderização em tempo real (segunda etapa), no intuito de obter acesso aos métodos de leitura e alteração de orientação e posição da câmera durante a pré-renderização. A Figura 2 ilustra o processo completo de renderização da cena.

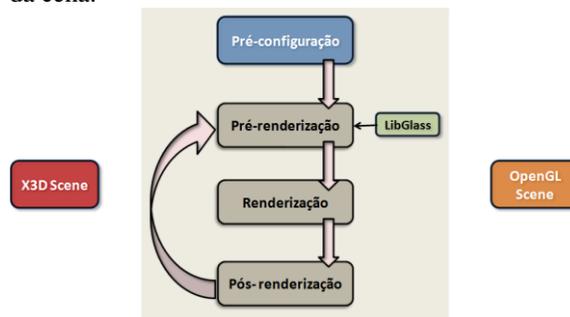


Figura 2 – Processo completo de renderização da cena.

O compartilhamento da visualização se dá através do padrão *peer-to-peer*, em que os nós mestres agem como um servidor, e os escravos como clientes. No mestre, as variáveis compartilhadas são atualizadas através do *input* do usuário, e informa aos nós escravos que de forma sincronizada acessam estes valores. Esta sincronização ocorre através das barreiras como o *datalock* e *framelock*.

O *datalock* é responsável pela coerência dos elementos dinâmicos da cena, e para isto é importante que as informações em cada nó sejam idênticas em cada instante. Essas informações são armazenadas nas variáveis compartilhadas ou ainda em eventos. Os dados são transmitidos para os outros nós para que cada um atualize sua posição e orientação coerente com a cena. A Glass força que todos os outros nós atualizem seus dados e sigam o nó mestre de acordo com a mudança ocorrida. Em seguida ocorre o *framelock* garantindo que alterações no ambiente ocorram ao mesmo tempo em todas as projeções. Sua ausência pode apresentar problemas de descontinuidade facilmente percebidos pelo usuário.

6. Resultados

Mediante os esforços da equipe e através do que já foi exposto, o objetivo de integrar a LibGlass e o Xj3D foi alcançado. Isto indica que qualquer cena X3D genérica pode ser executada em multiprojeção. Uma

vantagem deste trabalho em relação a [10] é que não foi preciso uma longa e complexa implementação de um *browser X3D* como o *Jinx*, além disso, o *Xj3D* é desenvolvido pela colaboração de diversos membros da comunidade científica e da *Web3D*, tornando-o estável e com mais funcionalidades suportadas. Contudo, nesta recompilação do *Xj3D* com a *LibGlass* apenas a clusterização da visualização foi implementada, ou seja, ambientes virtuais que possuam objetos interativos que envolva mudança de posição, tamanho, ou forma, ainda não são suportados nesta versão.

A Figura 3 permite observar o protótipo do *browser Xj3D* desenvolvido em execução, onde a visualização está distribuída em três telas, cada uma conectada a um computador diferente, interligadas em rede. Neste caso, as projeções estão configuradas para 60 graus de angulação. É possível navegar pelo ambiente sem a ocorrência de congelamento (travamento) da aplicação ou descontinuidade da cena. Os resultados podem ser replicados em mais computadores, configurados com diferentes angulações.



Figura 3 – Execução do *browser Xj3D* em multiprojeção.

7. Trabalhos Futuros

Como proposta para futuros trabalhos fica a implementação do compartilhamento de eventos no *Xj3D*. Alguns trabalhos [9] já a propõem por intermédio da API (*Application Programming Interface*) SAI (*Scene Access Interface*), mas o recurso não é implementado internamente ao *browser Xj3D*, tampouco é esta a maneira ideal para sistemas de multiprojeção, uma vez que barreiras imprescindíveis como o *framelock* não são utilizadas.

Outro fator importante quanto a sistemas de multiprojeção é a visualização estereoscópica no intuito de prover maior imersão ao usuário. Propõe-se ainda a implementação de visualização estereoscópica passiva através da polarização da luz, e uma interface gráfica para facilitar a utilização de estereoscopia, como por exemplo o ajuste de paralaxe.

8. Agradecimentos

Este trabalho está vinculado ao projeto Virtualidade Imersiva e Interativa baseada em *Cloud Computing* financiado pelo Ministério da Cultura através do edital XPTA.Labs, e por bolsas de pesquisa científica PIBIC através do CNPq.

9. Referências

- [1] Gnecco, B.B.; Guimarães, M.P. *Glass*. Laboratório de Sistemas Integrados, USP. São Paulo, Brazil. Outubro/2003.
- [2] Brutzman, D., Daly, L. “X3D: 3D Graphics for Web Authors”. Morgan Kaufmann Publishers. 2007.
- [3] Burdea, G.; Coiffet, P. “*Virtual Reality Technology*”. John Wiley e Sons, Nova Iorque, 1994.
- [4] Guimarães, M. P. Um ambiente para o Desenvolvimento de Aplicações de Realidade Virtual baseadas em Aglomerados Gráficos. Escola Politécnica da Universidade de São Paulo, São Paulo, 2004.
- [5] Jensen, C.; Pair, J; Flores, J.; Hodges, Larry F. The NAVE: Design and Implementation of Non-Expensive Immersive Virtual Environment. Gvu Technical Report; Georgia Institute of Technology, 2005.
- [6] Kumar, S.; Chhugani, J.; Kim, C.; Kim, D.; Kim, Y.; Nguyen, A.; Dubey, P.; Bienia, C. “*Second Life and the New Generation of Virtual Worlds*”, em *IEEE Computer*, Volume 41, Number 8. 2008.
- [7] Machado, L. S.; Costa, T. K. L.; Moraes, R. M. “*A 3D Intelligent Campus to Support Distance Learning*”, em *Proc. of Information Technology based Higher Education and Training (ITHET'2006)*, Sydney, 2006.
- [8] Marques, M.A.; Costa, T.K.L.; Machado, L.S.; Nettos, C.X.A. Representação do Sítio Arqueológico da Pedra de Ingá com Realidade Virtual. In: *Anais do Workshop de Realidade Virtual e Aumentada*. Santos, Brasil, 2009.
- [9] Sales, B.R.A.; Machado, L.S. Um Ambiente Virtual Colaborativo e Telecomandável Baseado em X3D. In: *Proc. X Symposium on Virtual and Augmented Reality (SVR2008)*. João Pessoa, Brasil, p. 327-330, 2008.
- [10] Soares, L. P. Um ambiente de multiprojeção totalmente imersivo baseado em aglomerados de computadores. Escola Politécnica da Universidade de São Paulo. São Paulo, 2005.
- [11] Web3D, “X3D Public Specifications”. Disponível em: www.web3d.org/x3d/specifications. Acesso em: março/2011.
- [12] X3DOM 1.0, “*Instant 3D the HTML way!*”. Online: <http://www.x3dom.org/>. Acesso em: março/2011.